

# SensLAB: a Very Large Scale Open Wireless Sensor Network Testbed<sup>\*</sup>

C. Burin des Rosiers<sup>2</sup>, G. Chelius<sup>2,1</sup>, T. Ducrocq<sup>2</sup>, E. Fleury<sup>1,2</sup>, A. Fraboulet<sup>3,2</sup>, A. Gallais<sup>4</sup>, N. Mitton<sup>2</sup>, T. Noël<sup>4</sup>, and J. Vandaële<sup>2</sup>

<sup>1</sup> ENS de Lyon – `firstName.lastName@ens-lyon.fr`

<sup>2</sup> INRIA – `firstName.lastName@inria.fr`

<sup>3</sup> INSA de Lyon – `firstName.lastName@insa-lyon.fr`

<sup>4</sup> Université de Strasbourg – `firstName.lastName@unistra.fr`

**Abstract.** This paper presents a precise description of SensLAB: Very Large Scale Open Wireless Sensor Network Testbed that has been developed and deployed in order to allow the evaluation through experimentations of scalable wireless sensor network protocols and applications. SensLAB’s main and most important goal is to offer an accurate open access multi-users scientific tool to support the design, development tuning, and experimentation of real large-scale sensor network applications. The SensLAB testbed is composed of 1024 nodes over 4 sites. Each location hosts 256 sensor nodes with specific characteristics in order to offer a wide spectrum of possibilities and heterogeneity. Within a given site, each one of the 256 nodes is able both to communicate via its radio interface to its neighbors and to be configured as a sink node to exchange data with any other sink node of the whole SensLAB testbed. The hardware and software architectures that allow to reserve, configure, deploy embedded software, boot wireless sensor nodes and gather experimental data and monitoring information are described in detail. We also present demonstration examples to illustrate the use of the SensLAB testbed.

**Keywords:** Wireless Sensor Network, Testbed, Radio, Monitoring, Experiments

## 1 Introduction

Wireless sensor networks (WSN) have emerged as a premier research topic. However, due to their massively distributed nature, the design, implementation, and evaluation of sensor network applications, middleware and communication protocols are difficult and time-consuming tasks. It appears strategic and crucial to offer to researchers and developers accurate software tools, physical large scale testbeds to benchmark and optimize their applications and services. Simulation is an important phase during the design and the provisioning step. However, they suffer from several imperfections as simulation makes artificial assumptions on radio propagation, traffic, failure patterns, and topologies [4, 7]. As proposed

---

<sup>\*</sup> Supported by ANR/VERSO program

by initiatives in Europe and worldwide [6], enabling “*open wireless multi-users experiment facility testbeds*”, will foster the emergence of the Future Internet and would be increasingly important for the research community. There is an increasing demand among researchers, industrials and production system architects to access testbed resources they need to conduct their experiments.

In this paper, we describe SensLAB, an open access multi-user WSN testbed, which has been created to answer all these needs. SensLAB provides appropriate tools, methods, experimental facilities for testing and managing their large scale wireless sensor network applications. As such, it lowers the entry cost to experimentation, often considered as a complex and heavyweight activity, with no extra management burden, accelerating proof-of-concept evaluation and competitiveness. In this paper, we first describe the hardware and software architectures of the platform, then we show how easy and efficient it is to use it through examples of application. The remainder of this document is organized as follows: Section 2 describes the general design requirements mandatory to insure the success of a large scale wireless sensor network testbed. We describe in Section 4 and Section 5 the various hardware and software modules and components developed within the SensLAB context. Section 6 presents some experimental results using the infrastructure and facilities offered by SensLAB.

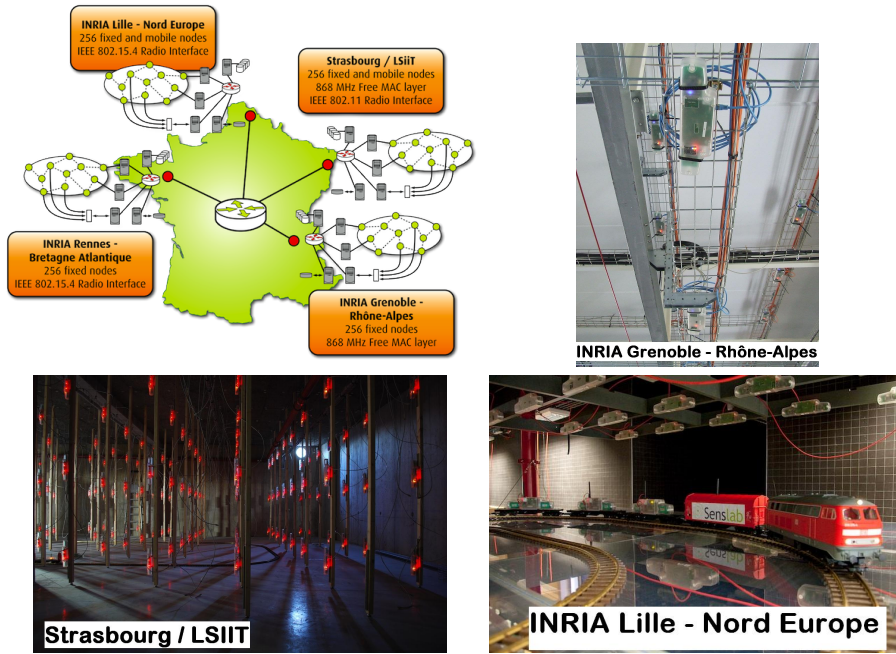
## 2 Context and Design Requirements

**Context.** In wireless sensor networks, a gap still remains in terms of experimental results which are very important to better understand the characteristics and compensate for some of these flaws and reach the state of maturity to make them practical. Unfortunately, the development and testing of real experiments quickly become a nightmare if the number of nodes exceeds a few dozens: *(i)* sensors are **small** devices with very **limited capacities** in terms of debugging and friendly programming; *(ii)* software deployment and debugging require the connection of the device **individual manipulation** of each single node; *(iii)* sensors are generally powered by a **battery** which has limited lifetime, etc. But developing and deploying a general purpose open wireless sensor network testbed, accessible to multiple users raises several challenges that are not present in wired network (Emulab [10], PlanetLab [8]) or even wireless mesh networks (Orbit [9]).

CitySense<sup>5</sup> consists of a set of nodes with high power radio deployed on rooftops and streetlights around Cambridge/UK. Currently there are 25 nodes deployed outside and another 32 nodes deployed as part of an indoor testbed. It does not allow testing application on small devices constrained by energy, memory, CPU, etc. ORBIT [9] focuses on the creation of a large-scale wireless network testbed to facilitate a broad range of experimental research on next-generation protocols and application concepts. The ORBIT’s philosophy is similar to SensLAB but it is dedicated to IEEE 802.11-based networks.

---

<sup>5</sup> <http://www.citysense.net/>



**Fig. 1.** SensLAB testbed is composed of 4 distributed WSN interconnected by Internet (1024 nodes in total). Mobile nodes are available on Strasbourg and Lille thanks to several toy electric trains remotely controllable and to supply power to the nodes.

There exist other WSN testbeds like moteLab<sup>6</sup>, Kansei<sup>7</sup>, WASAL<sup>8</sup> or TWIST [3] but smaller and not targeting the same objectives as SensLAB. In some of them, specific operating system (i.e. TinyOS), MAC layer (i.e. IEEE 802.15.4) or/and program language (i.e. NesC) are imposed, which limits some optimizations. Usually, these platforms do not provide radio instrumentation and/or noise injection. Recently, the WISEBED<sup>9</sup> project shows the ambition to federate several WSN testbeds in order to establish a European wireless sensor network. It seems that application should be developed by using a specific API dedicated to the WISEBED platform. A great benefit of the WISEBED project is the release of Wiselib, an algorithm library for sensor networks (localization, routing).

**SensLAB functionalities.** We propose to eliminate some of the problems listed above by operating SensLAB as an *open* research facility for academic and industrial groups. SensLAB provides a research infrastructure for networking sensors for the exploration of sensor network issues in reproducible experimental condi-

<sup>6</sup> <http://motelab.eecs.harvard.edu/>

<sup>7</sup> <http://ceti.cse.ohio-state.edu/kansei/>

<sup>8</sup> <http://wasal.epfl.ch/>

<sup>9</sup> <http://www.wisebed.eu/>

tions. The platform is **generic**, **open** and **flexible**: it means that a user is able to remotely (web access) access and deploy his/her applications *without any kind of restrictions on the programming language, on the programming model or on the OS that he/she desires to use*. The testbed should be **scalable**, cover a large spectrum of sensor network applications and target several kinds of end users.

SensLAB answers all these requirements, providing an easy way to set an experimentation, letting the user chooses the node number, sensor & radio characteristics, topology considerations, experimentation time, etc. SensLAB integrates an efficient reservation tool to schedule different experimentations. The user can keep an on-line access to his/her nodes (either by the web or by a command line shell) at anytime.

To reach the main objectives of the testbed, strong efforts and developments have been given both on the *hardware* and *software* sides. The hardware design of the SensLAB nodes must satisfy several strong requirements: *(i)* **reliable access** to the nodes in order to perform operations such as a reset or code flashing whatever the state of the sensor node or the software it is running. *(ii)* non intrusive and application transparent **real time monitoring** of the sensor nodes. The external monitoring (*i.e.*, totally independent from the deployed user application code) will include precise and real-time access to fundamental parameters such as energy consumption and radio activity on a per node basis; *(iii)* **security and data integrity** between consecutive experiments on the same set of nodes; *(iv)* **real time control** of the experiments by providing a set of commands that may influence an application environment (e.g., turn sensor nodes on/off to mimic crashes, emit radio noise by sending fake data in order to tamper with transmissions, modify the monitoring frequency parameters); *(v)* if experimental results / validations are important, they should be **reproducible**.

### 3 Main Elements of SensLAB

Fig. 2 gives an overview of the testbed services. The user sets his/her experiment through a webportal. A virtual machine is setup with all the development tools and chains preconfigured (cross compilation chains, OS, drivers, communication libraries). The user can also access and use higher level development and prototyping tools (like cycle accurate hardware platform simulator [2] or more conventional radio accurate wireless network simulator [1]). When an experiment is launched, specific SQL tables related to the experiment are created. All monitoring data collected during an experiment are stored in tables to support subsequent analysis. The user is thus able to perform postmortem analysis but the system also provides online data analysis services (OLAP/ On-Line analytical Processing). We will describe more precisely all the different elements of SensLAB in the two following sections. Each service is replicated on each site in order to be fault tolerant (DNS for users virtual machines, LDAP for authentication...). Fig. 3 details all the software components deployed on each site.

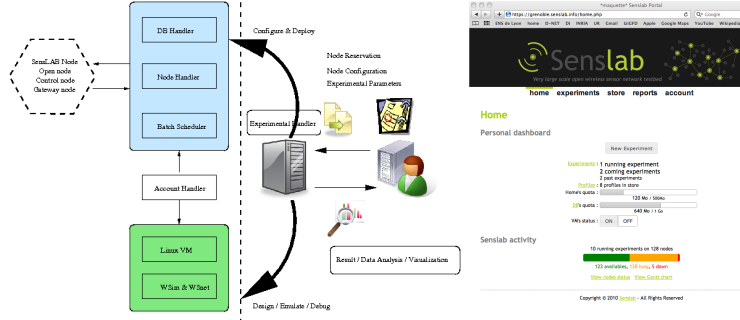


Fig. 2. Simplified view of the platform usage/services (left). SensLAB portal (right).

## 4 SensLAB Hardware Components and Infrastructures

All the requirements listed in Section 2 have a strong impact on the hardware and on the architecture needed to support the testbed and handle all user experiments. More precisely, the SensLAB hardware infrastructure consists of three main components: (i) The **open wireless sensor node** dedicated to the user, (ii) The full **SensLAB node** that encompasses the open node also includes a gateway and a closed wireless node, (iii) The global networking backbone that provides power and connectivity to all SensLAB nodes and guarantees the out of band signal network needed for command purposes and monitoring feedback.

### 4.1 Open Wireless Sensor Node

The **open wireless sensor node** is made available to the user during his/her experimentation. This node is totally open and the user is granted a full access to the memory. The choice of the sensor node is a crucial task since it is related to the broad range of wireless sensor network problems that might be anticipated over the next 4-8 years. The current trend for wireless sensors nodes is geared toward a common architecture based on off the shelf 16-bit micro-controllers. We thus clearly target low power wireless sensor nodes constrained in memory and energy like existing products already on the market<sup>10</sup>. In order to meet with the requirements in terms of energy monitoring, reproducibility, we need to master the architecture and a solution has been to design our own board in order to include all control signals and thus guarantee a reliable control and feedback<sup>11</sup>. The nodes are based on a low power MSP430-based platform, with a fully functional ISM radio interface<sup>12</sup>.

<sup>10</sup> WiEye, Micaz, Tmote-Sky, TinyNodes

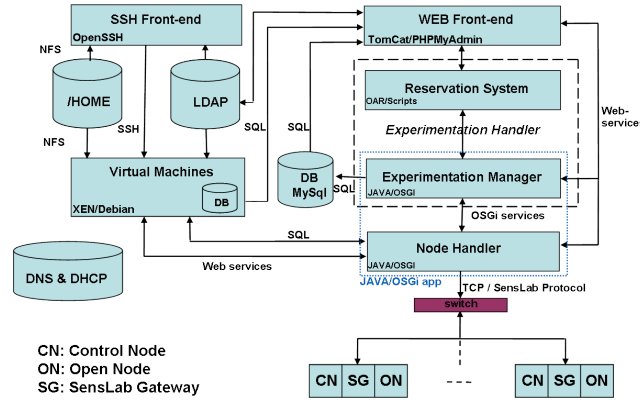
<sup>11</sup> All designs are released under a Creative Commons License

<sup>12</sup> Two versions are currently available: an open 868MHz radio interface and IEEE 802.15.4 radio interface at 2.4GHz.

## 4.2 One SensLAB node: 2 WSN430 and a Gateway

As specified earlier, a WSN430 node itself is more or less useless if it is not equipped with a reliable way to control it (energy supply, code deployment, monitoring). To control the *open* WSN430 node that the user will request and use, we choose to mirror it with another WSN430 whose specific role is to control the open one. In order to link the two WSN430 nodes and also to meet with all mandatory requirements listed previously, we design the SensLAB gateway board that plays a key role in the control and management of the platform: Automated firmware deployment on open node; Accurate power monitoring of open nodes, both on battery and DC power supply Expected measure precision is 10uA, and power sampling around 1kHz.; Radio environment monitoring control, (RSSI measures and noise injection), thanks to the control node; Configurable sensor polling on the control node (temperature, light, acoustic activity); Fixed (Ethernet) as well as mobile (Wifi) communication with Node Handler via a Gigi Connect module; Power over Ethernet support for a standardized and easy power management; Sink capability for each open node (in and out characters stream redirection); Option for daughter cards on open and control node; Remote firmware update ability for control the node and the gateway. A scheme of the Gateway connection is available at **TODO: ADD URL where FIG Senlab gateway is available**.

## 5 SensLAB Software Architecture



**Fig. 3.** Software SensLAB architecture and technological choices.

The SensLAB software architecture is replicated over the four testbed sites, and it is divided in several parts which interact together as shown by Fig. 3. We detail below their functionalities.

**Control node software:** i.e. the firmware running on the control node. It is in charge of powering up/down, resetting and doing measures on the open node

activity (power consumption, radio activity/RSSI). It can set on its two DAC (digital to analog) pins any voltage between ground and power supply. Through those pins connected to the open node's ADC (analog to digital converter), the control node can send stimuli to open node. All these actions can be executed either asynchronously on the user's request, or automatically and periodically.

**Gateway node software:** i.e. the firmware running on the gateway. It manages the interface between the open and control nodes, and the SensLAB site server over IP communications. It forwards the command frames addressed to the control node, updates the open node's firmware (BSL protocol), and forwards the open node's serial link to the server (sink application).

**Experiment handler software:** The experiment handler software (Fig. 4) is the server side interaction point with all the 256 nodes. It can execute all methods described above such as firmware update, energy consumption monitoring, polling, etc. It also receives the data coming from the serial links of open nodes relayed by the gateway nodes. It instantiates a 'testbed manager' object and an OSGi container when started. The OSGi container embeds several bundles, responsible for all the interactions with the nodes: the Node Handler bundle sends command frames to the gateway and the control node; the Firmware Deployer bundle provides one service allowing parallel deployment of a firmware on several open nodes; the Sensor Controller bundle allows parallel sensor measurement; the Sink Forwarder bundle provides efficient data redirection between nodes and users' VMs.

**Batch scheduler software:** Through a web form (or by uploading an xml file), the user configures his/her experiment and specifies wanted nodes (either mobile or fixed nodes, with a CC1100 or CC2420 radio chip, outdoor or indoor, location and the number of nodes), experiment's duration and eventually a start date. Those information are transmitted to the batch scheduler software, which is the server-side module allowing optimal experiments scheduling and resources allocation. It is also in charge of triggering start and stop of planned experiments, by invoking the experiment handler for nodes configuration and resources release. This module is based on the use of OAR<sup>13</sup>, which is a versatile resource manager (or batch scheduler) for large clusters.

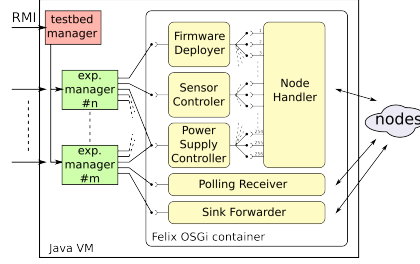
**User virtual machines:** A complete Linux environment is made available to each registered user allowing him/her to build sensor firmwares thanks to the complete set of tools installed, interact with the nodes of his/her running experiment and running a dedicated application to handle the nodes' serial link outputs. This client, an interactive prompt, connects to the Experiment Handler software and performs firmware update, power supply modification and asynchronous measures on any of the experiment's nodes.

## 6 Conducting experimentations with SensLab

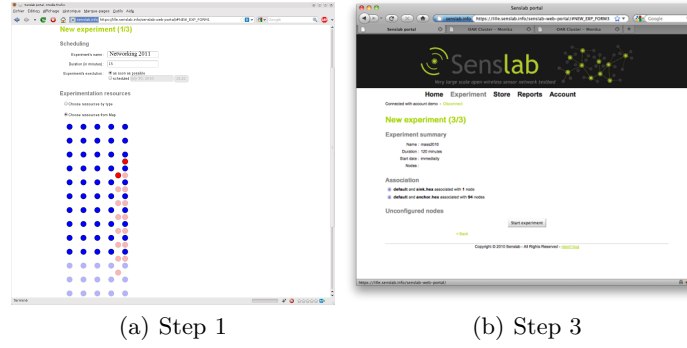
When a user wishes to conduct some experiments on the SensLAB platform, he/she first needs to configure his/her experiment through the webportal. As

---

<sup>13</sup> <http://oar.imag.fr>



**Fig. 4.** Experiment Handler software structure.



**Fig. 5.** The SensLAB web portal for Lille’s platform.

displayed by Fig. 5(a), SensLAB offers an easy way to select nodes that we need. Through this web portal page: the user is able to choose the experiment name (*Networking 2011* on Fig.), the duration of the experiment (15min) and if the experiment has to start as soon as possible or at a chosen date (*as soon as possible* in our case). Then, nodes on which the experiment will be run have to be chosen, either through their Id or on the map. These nodes are then associated with a specific code/firmware to be tested that the user uploads. A summary page is then displayed (Fig. 5(b)). In our case, one node is configured as a sink and 94 as anchors.

When configuration is done, the experiment will be submitted. SensLAB testbed automatically reserves nodes. At starting time, *i.e.* in our case, as soon as chosen nodes are all available, SensLAB configures *control nodes*, deploys firmwares, and resets the experiment’s nodes. Every action is performed automatically and in a transparent way for the user.

Once the experimentation is launched, the user is able to interact with the running experiment as we will see in following sections illustrating an experiment. Results chosen to be retrieved (by polling or reading on serial port) are available in the user’s virtual machine.

In the remaining of this section, we give some examples of utilization of the platform through several applications examples.



## 6.1 SensLab for testing geolocalization protocols

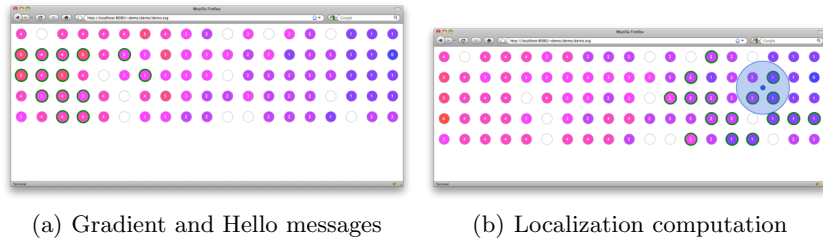
To illustrate the benefits and use of the SensLAB testbed in designing new algorithms, we focus on an animal tracking application [5]. Indeed, biologists need to track some animals to learn from their way of life (especially in natural parks). To do so, animals are instrumented with sensors. They are mobile and their tracks need to be followed. To geolocalize them, some fixed nodes called *anchors* are spread in the park and receive signals from mobile nodes as soon as they are in range. *Anchor* nodes register the mobile node identifier, the RSSI (Received Signal Strength Indicator) of the signal and the date. Then, these data needs to be routed to a *sink* node. This latter is connected to a computer gathering data and computing *mobile* node location based on these data. Note that geolocalization application has been simplified as possible since the main purpose here is to highlight SensLAB benefits.

In the setup of such an application, the first step is to cover the bounded area by deploying *anchors*. Next step is to set up the routing infrastructure to allow every anchor to send data to the sink. Once again, the routing process chosen here is simplified. When *anchors* are deployed and powered on, the *sink* is initialized. It then starts to send BEACON and every *anchor* receiving this BEACON attaches itself to the sink. The *sink* becomes its parent. Then every attached *anchor* *a* forwards the BEACON. Every unattached *anchor* receiving a beacon from *a* chooses *a* as its parent. When every *anchor* has chosen a parent, the whole area is covered and *mobile* messages can be forwarded to the *sink* as follows. When an anchor receives a data message from another anchor or needs to send its own data, it forwards it to its parent. Step by step, the message eventually reaches the *sink*. The *sink* sends data through its serial link and the computer connected to it gathers the different messages and estimates mobile node positions.

**Demonstration overview:** Demonstration will be conducted from the Lille's SensLAB site which is located in the north of France. Nodes of this site are featured with CC2420 radio chip and 32 among the 256 nodes are mobile, mounted on several toy electric trains. For the need of the application, we will reserve a grid of  $5 \times 10$  nodes and 2 mobile nodes located on different train paths. Mobile nodes will represent the animal while fixed nodes will stand for anchor nodes. The reservation will be performed through the SensLAB web portal (see Section 3). Three firmwares have to be compiled for the experimentation, respectively for *mobile*, *anchor* and *sink* nodes. In the experiment only one *profile* will be used tuned with nodes on DC power. Consumption polling is sampled every 50 ms. All fixed nodes except one will be associated with *anchor* firmware and the last node is associated with the *sink* firmware.

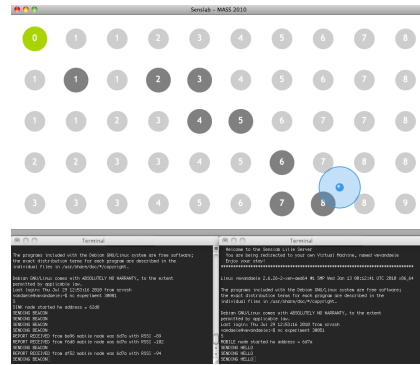
Experimentation is launched. During the experiment, monitoring data (power consumption, RSSI, ...) are logged like described in the experiment profile which is the same for all nodes. Once the experimentation is launched, the user is able to interact with the running experiment via his/her VM as shown by Fig. 7. Interactions with the nodes are made by using the user's VM through an SSH connection. From the VM, and using the UNIX *netcat* tool to create and open a

TCP socket on a port of a special host, which is the serial link redirection of the node, the *sink* is ready to be activated and to start sending BEACONS. Results are gathered in real time. A route tree is created by the *anchors* to forward reports to the *sink* (In Fig. 6(a), numbers on nodes give the distance each node estimates itself to the sink at a given time.). *Mobile* nodes are activated through their serial links and start sending Hello Messages received by anchor nodes (Green nodes in Fig. 6(a)). Based on the RSSI of these Hello messages, anchors compute the location of mobile nodes (Fig. 6(b)).



**Fig. 6.** Real time interaction with the running experiment.

For the experimentation purpose, the VM is hosting an application which collects data from serial links, analyzes them to compute *mobile* node locations and provides a web server to visualize application status in real time (Fig. 7). *Anchors*, routes, messages and estimated *mobile* node positions can then be displayed in a web browser.



**Fig. 7.** Demo visualization interface showing routes, messages transmission and mobile nodes location in the upper window, and messages printing on serial links for sink and mobile nodes on lower windows.

## 6.2 Strasbourg applications

**TODO: TO be completed**

## 7 Conclusion

The architecture concepts, the hardware design, the software implementation of SensLAB, a large scale distributed open access sensor network testbed were presented. We have also provided examples showing how to use SensLAB to launch and evaluate experiments. The testbed is deployed and operational and under beta test and counts **TODO: COMPLETE**

XX registered users. We are working on the deployment of the OAR-grid version in order to allow fully flexible distributed node reservation. Several research works remain. One direction concerns the study of the federation of research platforms, and more precisely with OneLab. **TODO: On cite FLAB ?**

A federation will offer a higher dimension in the spectrum of applications that the research community will design, test, deploy, and tune. But even more important, SensLAB will strongly benefit from the monitoring tools and supervising infrastructure developed and used in OneLab. Other extensions concern the use of hybrid simulation within SensLAB. The last extension is the development of actuator nodes, plugged directly on SensLAB nodes.

## References

1. E. Ben Hamida, G. Chelius, and J.-M. Gorce. Impact of the physical layer modeling on the accuracy and scalability of wireless network simulation. *Simulation*, 85(9):574–588, 2009.
2. G. Chelius, A. Fraboulet, and E. Fleury. Worldsens: development and prototyping tools for application specific wireless sensors networks. In ACM, editor, *International Conference on Information Processing in Sensor Networks (IPSN)*, 2007.
3. V. Handziski, A. Köpke, A. Willig, and A. Wolisz. Twist: A scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *RealMAN 2006*, 2006.
4. S. Kurkowski, T. Camp, and M. Colagrosso. Manet simulation studies: the incredibles. *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(4):50–61, 2005.
5. N. Mitton, T. Razafindralambo, and D. Simplot-Ryl. *Theoretical Aspects of Distributed Computing in Sensor Networks*, chapter Position-Based Routing in Wireless Ad Hoc and Sensor Networks. Springer, 2010. To appear.
6. NSF workshop on network research testbeds, 2002.
7. K. Pawlikowski and J. L. R. Jeong. On credibility of simulation studies of telecommunication networks. *IEEE Communications Magazine*, 2001.
8. L. Peterson, A. Bavier, M. Fiuczynski, and S. Muir. Experiences implementing planetlab. In *OSDI*, 2006.
9. D. Raychaudhuri, M. Ott, and I. Secker. Orbit radio grid tested for evaluation of next-generation wireless network protocols. In *TRIDENTCOM*, 2005.
10. B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. In *OSDI'02*, pages 255–270, 2002.